What is claimed is:

1. A method for scheduling data flows among processors, comprising,

receiving a request for processing,

identifying a processor group to process the request, the processor group including at least one processor;

consulting a flow schedule associated with the identified processor group, and,

transferring the request to at least one processor in the identified processor group based on the associated flow schedule.

2. A method according to claim 1, wherein receiving a request for processing includes receiving a data flow from a network.

3. A method according to claim 1, wherein consulting a flow schedule further comprises consulting a flow schedule vector.

4. A method according to claim 1, wherein transferring the request includes transferring the request based on a sequentially moving among processors in the consulted flow schedule.

5. A method according to claim 4, wherein sequentially moving among processors includes returning to the beginning of the

consulted flow schedule upon reaching the end of the consulted flow schedule.

6. A method according to claim 1, further comprising computing a flow schedule based on intrinsic data from the identified processor group.

7. A method according to claim 6, wherein computing a flow schedule based on intrinsic data includes computing a flow schedule based on at least one of CPU utilization, memory utilization, packet loss, and queue length or buffer occupation of the processors in the identified processor group.

8. A method according to claim 6, wherein computing a flow schedule further comprises receiving the intrinsic data from processors in the identified processor group.

9. A method according to claim 8, wherein receiving data from processors further includes receiving data at specified intervals.

10. A method according to claim 6, wherein computing a flow schedule further comprises filtering the intrinsic data.

11. A method according to claim 1, further comprising providing processor groups, the processor groups having at least one processor and wherein the processors in a processor group include at least one similar application.

12. A method according to claim 1, further comprising providing processor groups, the processor groups having at least one processor and wherein the processors in a processor group are identically configured.

13. A method according to claim 12, further comprising computing a flow schedule for the processor groups.

14. A method according to claim 1, further comprising providing processor groups wherein the processors in different processor groups include at least one different application.

15. A method according to claim 1, wherein consulting a flow schedule further includes providing an initial flow schedule.

16. A method according to claim 1, wherein identifying a processor group includes identifying an application associated with the request.

17. A method according to claim 1, wherein identifying a processor group includes consulting a hash table.

18. An apparatus to process a data flow on a network, comprising,

at least one flow processor module having at least one processor,

at least one network processor module having at least one processor, at least one interface to receive the data flow from the network, and instructions to cause the at least one processor to forward the data flow to at least one flow processor module capable of processing the data flow, and,

at least one control processor module in communication with the at least one flow processor module, and having at least one processor and instructions for causing the at least one processor to receive intrinsic data from the at least one flow processor module.

19. An apparatus according to claim 18, wherein the at least one flow processor module includes at least one memory to store at least one application.

20. An apparatus according to claim 18, wherein the at least one control processor module is in communication with the at least one network processor module.

21. An apparatus according to claim 18, wherein the at least one control processor module includes instructions for causing the at least one processor to compute a flow schedule for the at least one applications processor group.

22. An apparatus according to claim 18, wherein the intrinsic data includes at least one of CPU utilization, memory utilization, packet loss, and queue length or buffer occupation.

23. An apparatus according to claim 18, wherein the control processor modules further include at least one filtering module.

24. An apparatus according to claim 18, wherein the network processor modules further include at least one flow schedule for directing flows to the flow processor modules.

25. An apparatus according to claim 18, wherein the network processor modules further include at least one initial flow schedule.

26. An apparatus according to claim 18, wherein the network processor modules further include a hash table to associate the data request with a flow schedule.

27. An apparatus according to claim 24, wherein the flow schedule further includes a list of flow processor modules.

28. An apparatus for scheduling data flows on a network, comprising

a front-end processor to receive data flows from the network,

at least one applications processor group to process the flows,

at least one flow schedule associated with the at least one applications processor group, and,

instructions to cause the front-end processor to identify at least one applications processor group to process the flow, select at least one processor within the identified processor group, and transfer the flow to the selected processor.

29. An apparatus according to claim 28, wherein the at least one flow schedule includes at least one flow vector.

30. An apparatus according to claim 28, further comprising at least one control processor to receive data from the at least one applications processor group.

31. An apparatus according to claim 30, wherein the control processor includes at least one filter.

32. An apparatus according to claim 28, wherein the at least one applications processor group includes at least one processor.

33. An apparatus according to claim 32, wherein the at least one processor includes at least one memory to store applications.

34. An apparatus according to claim 28, wherein the front-end processor includes a hash table for associating a data flow with at least one applications processor group.

35. A method for scheduling data flows among at least two processors, comprising computing a flow schedule based on historic performance data from the at least two processors.

36. A method according to claim 35, wherein computing a flow schedule based on historic performance data includes providing historic data for at least one of CPU utilization, memory utilization, packet loss, and queue length or buffer occupation of the processors in the identified processor group.

37. A method according to claim 35, wherein computing a flow schedule based on historic performance data includes providing presently existing data for at least one of CPU utilization, memory utilization, packet loss, and queue length or buffer occupation of the processors in the identified processor group.